用户兴趣变化和类别关联度的混合推荐算法 *

陈海龙，谢　晟，薛宇彤

(哈尔滨理工大学 计算机科学与技术学院, 哈尔滨 150080)

摘　要：协同过滤算法是目前推荐系统中最普遍的个性化推荐技术。针对传统算法相似性度量方法不足的问题，提出了融合用户兴趣变化和类别关联度的混合推荐算法。算法根据用户的评分项目信息来对项目进行类别划分，挖掘出用户对不同类别项目的喜爱关注程度；同时将基于时间的兴趣度权重函数引入项目相似度计算之中来进一步提高计算的精确度，最后将改进后的相似度计算方法融入到用户聚类方法中，用户聚类之后，其所在的类别将对用户推荐准确度产生极大的作用。实验结果表明，在 Movielens-1k 数据集上运行该算法，该算法在运行效率和精确度上都有所提高。

关键词：协同过滤；聚类算法；类别关联度；兴趣变化；相似度

中图分类号：TP301.6　　doi: 10.3969/j.issn.1001-3695.2017.08.0722

Hybrid recommendation algorithm for user interest change and category related degrees

Cheng Hailong, Xie Sheng, Xue Yutong

(*School of Computer Science & Technology*, *Harbin University of Science & Technology*, *Harbin* 150080, *China*)

Abstract: Recommendation system has been widely applied to various types of e-commerce sites, which effectively solved the problem of information overload, collaborative filtering algorithm is the most common in the recommendation system of personalized recommendation technology. Based on the problem of the traditional method of similarity measurement, a hybrid recommendation algorithm is proposed to combine the change of interest and class correlation degree. The algorithm classifies the project according to the user's rating project information, and finds out how much the user likes to pay attention to different categories of projects. At the same time, the time based interest weight function is introduced into the project similarity calculation to further improve the accuracy of calculation. Finally, the improved similarity calculation method is integrated into the user clustering method. After the user clustering, the category of its location will have a great effect on the user's recommended accuracy. The experimental results show that the algorithm is improved in operation efficiency and accuracy in the moviels-1k data set.

Key Words: collaborative filtering; clustering algorithm; class correlation; interest change; similarity

0　引言

随着互联网规模和覆盖面的不断拓宽，网上信息数据以爆炸式速度迅速增长，过量的信息同时出现在用户面前使得用户无法从中分辨和获取有效的信息，信息利用率低下，造成信息超载。推荐系统是当前解决信息超载问题的非常有效的方法，推荐系统根据用户的信息需求、兴趣等，将用户感兴趣的信息、产品等推荐给用户。和搜索引擎相比推荐系统通过研究用户的兴趣偏好，进行个性化计算，由系统发现用户的兴趣点，从而引导用户发现自己的信息需求。基于用户行为数据分析的推荐算法称为协同过滤算法，其基本思想是具有相似行为的用户之间具有相似的需求爱好。因此协同过滤算法更关注用户的历史行为，不受新项目的影响，具有更好的推荐精度。

为了解决传统协同过滤算法推荐精度不高以及数据集稀疏等问题，许多学者提出了相似度改进算法以及其他的算法如聚类算法。例如：宋瑞平提出了基于用户评分及用户属性的相似度计算方法和改进的最近 k 邻的混合推荐算法—MSCF 算法[1]，提高了推荐算法的精确度；以上方法进一步提高了算法的准确性，但在计算用户相似度时，仅仅考虑了用户评分数据，并没有考虑用户的共同评分即用户的评分差异度，忽略了项目的类别喜爱度以及类别关注度等问题。为了解决这些问题，相关学者还引入聚类技术对协同过滤算法进行优化。如尹航提出的采用聚类算法优化的 k 近邻协同过滤算法[3]。 用户相似性度量除了考虑用户对项目的评分数值，与用户的兴趣也有很大的关联。文献[7]通过加入用户的信任度和项目属性信息，利用基于遗忘规律的兴趣变化时间策略对用户进行近邻集合的推荐。

基金项目：黑龙江省自然科学基金资助项目（A201301）；哈尔滨市科技创新人才研究专项资金资助项目（RC2017QN010029）

作者简介：陈海龙（1975-），男，教授，硕导，博士，主要研究方向为知识工程、时间序列；谢晟（1994-），男，硕士研究生，主要研究方向为推荐算法；薛宇彤（1994-），女，硕士研究生，主要研究方向为推荐算法．

针对以上问题，本文提出了一种基于用户兴趣变化和类别关联度的聚类协同过滤算法。用户的兴趣可能会随着时间变化，因此将兴趣变化曲线融入到项目相似度计算当中。类别关联度即类别喜爱度或类别关注度，代表着一定程度上的用户偏好。基于以上两个因素，本论文首先将对项目进行聚类，然后根据项目聚类结果进行改进相似度计算，再根据相似度对用户进行聚类，这大大降低了聚类的时间复杂度，同时对数据信息充分的应用了。

1　传统的协同过滤推荐算法

基于用户行为数据分析的推荐算法称为协同过滤算法，其基本思想是具有相似行为的用户之间具有相似的需求爱好。因此协同过滤算法更关注用户的历史行为，不受新项目的影响，具有更好的推荐精度。基于协同过滤的算法主要有基于用户（项目）的推荐算法、基于模型的推荐算法以及混合推荐算法。基于内存的协同过滤算法包括基于用户的方法和基于项目的方法，主要分为三个步骤：基于用户-项目评分矩阵，计算用户（项目）之间的相似性；通过相似度的逆序，选取最相似的前K 个用户（项目）作为邻居；根据邻居的评分，对目标用户（项目）未评分的项进行预测。下面以基于用户的协同过滤算法为例进行详细说明。

1.1　用户—项目评分模型

定义一个给定的用户集 U 和项目集 S，用户对项目的评分表示为一个 m×n 的矩阵 R，如表 1 所示。R（i，j）表示用户 i 对项目 j 的评分， 代表用户对项目的偏好。如 MovieLens 数据集中用 1~5 分表示用户的喜爱程度；若 R（i，j）=0 则表示用户 i 未对项目 j 打分。

表 1　用户—项目 m×n 阶评分矩阵 R

	S_1	…	S_j	…	S_n
U_1	$R_{1,1}$	…	$R_{1,j}$	…	$R_{1,n}$
…	…	…	…	…	…
U_i	$R_{i,1}$	…	$R_{i,j}$	…	$R_{i,n}$
…	…	…	…	…	…
U_m	$R_{m,1}$	…	$R_{m,j}$	…	$R_{m,n}$

1.2　用户间相似性度量公式

1）余弦相似度

基于以上用户—项目评分矩阵来进行用户相似度的计算，用户相似度计算方法有余弦相似度算法和皮尔森算法，这里采用的是余弦相似度算法，将用户评分看做一个 n 维的评分向量，第 k 维的值表示对项目 k 的评分，设用户 u 与用户 v 的评分向量分别表示向量 u 和 v，则用户 u 和用户 v 之间的相似度为

$$sim(u,v)=\cos(\vec{u},\vec{v})=\frac{\vec{u}\bullet\vec{v}}{\|\vec{u}\|\bullet\|\vec{v}\|} \tag{1}$$

相似度范围[-1,1]，值越大，则用户 u 和 v 兴趣爱好越接近。

2）相关相似性（pearson 相关系数）

余弦相似性的度量方法存在一定的精确度问题，并未考虑到不同用户的评分尺度问题，所以给出 pearson 相关系数。定义两用户 u 和 v 共同评分过的项目集合为 I(u,v),其中 I(u)和 I(v)分别为用户 u 和用户 v 评分过的项目集合，他们的平均评分为 r(u)、r(v)，用户 u 对项目 s 的评分为 r(u,s)，则相似度为 sim(u,v)为

$$sim(u,v)=\frac{\sum_{s\in I(u,v)}(r(u,s)-r(u))*(r(v,s)-r(v))}{\sqrt{\sum_{s\in I(u,v)}(r(u,s)-r(u))^2}*\sqrt{\sum_{s\in I(u,v)}(r(v,s)-r(v))^2}} \tag{2}$$

相似度范围[-1,1]，值越大，则用户 u 和 v 兴趣爱好越接近，本论文使用 pearson 相关系数求解相似度。

计算相似度之后，选取与用户 u 相似度最大的前 k 个用户作为邻居 G(U)，依据这 k 个用户对目标项目 j 的评分，加权预测用户 u 对项目 j 的评分 $P_{u,j}$，如下公式所示，其中 $\bar{R}_u$ 表示用户 u 对评分的均值，$R_{u,j}$ 表示用户对 u 项目 j 的评分。

$$P_{u,j}=\bar{R}_u+\frac{\sum_{i\in G(u)}^{n} sim(u,j)(R_{u,j}-\bar{R}_u)}{\sum_{i\in G(u)}^{n} sim(u,j)} \tag{3}$$

2　基于时间的兴趣度权重

为用户进行推荐，着重观察他的评价数据，并从中挖掘出他的兴趣爱好以及需求，最后为他推荐相关产品。实验数据集中记录了每位用户对每部电影评价时的具体时间，因此根据这个数据特点，本文可以充分运用数据，发掘该用户的近期喜好变化。人们对不同电影类别以及电影的喜爱会随着时间产生变化，并且用户近期观看过的电影更能为本文预测其未来感兴趣的资源。受遗忘规律启发，参考 Ebbinghaus 遗忘曲线函数的特征，设 s（u,i）为用户 u 对项目 i 的兴趣度。考虑到用户对项目的评价时间有前有后，设 t0 为用户 u 对项目评分的最早时刻，t(i)为用户对项目 i 的评价时刻，则 s（u,i）可以表示为：

$$s(u,i)=\frac{e^{-(ti-t0)}}{ti-t0} \tag{4}$$

如果 t0=ti,则定义 s(u,i)=1。

每个用户的兴趣变化速度与规律不同，用户兴趣也存在各种反复以及变化，因此对于用户早期的访问数据，也应该重视并充分利用，以下为过往用户兴趣相似度的度量函数 I（u,i），设用户 u 已访问的项目集合为 I(u)，定义一时间段 T，用户 u 在最近 T 时间段访问过的项目集合为 I（u,t），I（u,t）表明了用户的近期兴趣，对于一个项目，如果 u 访问的近期项目集合中很多项目都与 i 相似度很高，则说明项目 i 与用户当前兴趣具有很大关联，未来用户的兴趣可能还与项目 i 相似。所以项目 i 在预测用户兴趣时起到关键作用。

通过 i 与 I（u,t）中的项目总体相似度计算 I（u,i）：

$$I(u,i)=\frac{\sum_{j\in I(u,t)} sim(i,j)}{size(I(u,t))} \tag{5}$$

其中:size(I（u,t）)为用户 u 在最近 T 时间段内访问过的资源集合大小。

由以上分析可知，结合基于用户近期兴趣变化以及用户远期兴趣数据来对数据进行分析操作是很有必要的。兴趣变化频繁的用户更注重近期的兴趣，近期喜爱项目的比重要大于远期项目，而基于过往用户兴趣度量函数对远期数据进行操作，则是为了将数据进行充分的运用，避免遗漏早期关键数据的特点，无法真正把握用户兴趣存在反复的情况，最后，将近期用户兴趣的度量函数与远期用户兴趣度量函数相结合得

$$f(u,i)=\alpha\times s(u,i)+(1-\alpha)\times I(u,i) \tag{6}$$

3 相似度改进的用户聚类协同过滤算法

本论文将采用K均值聚类方法进行用户和项目聚类，传统的聚类算法步骤为:

输入：K 个用户分类。

a)从用户集合 U 中随机取 K 个用户，作为 K 个簇各自的中心。

b)分别计算用户 UI 与 K 个簇中心的相似度，将 UI 归到相似度最大的类别。

c)重新计算 K 个簇各自的中心，计算方法是求出所有用户对项目的评分算术平均值，作为簇中心点。

d)对于 U 中所有用户重复步骤 b)c)的迭代法更新，直到聚类结果保持不变，则迭代结束，否则继续迭代。

基于类别关联度的相似度计算

对用户进行相似度计算，确定簇中心时，本文对相似度的计算定义使用的是 pearson 系数，相对而言，计算比较准确，但是每个用户对电影的偏好不同，且近期用户兴趣也会对相似度计算结果产生影响，因此对项目计算相似度计算时，本文应该考虑多方面的影响因素。

完成项目聚类操作之后（按照数据集定义的类别来进行聚类），项目被聚集为K类，每一类中项目都具有相似性质。每个用户有不同的偏好，因此当项目进行聚类之后，本文就可以获得用户对类别的喜爱程度了。当用户进入网站进行电影选择时，不知道影片的具体内容而是只知道该影片的类别，用户也会有一定的偏向性，即使当他观看完该电影之后，并不喜欢这不电影，因此定义用户 I 对电影类别 J 的类别喜爱度为 fav(i,j):

$$fav(i,j)=\frac{\sum_{k\in c(j)} r(i,k)}{\sum_{I\in c} r(i,I)} \tag{7}$$

根据以上公式，可以求出用户 I 和 J 之间的聚类相似度 simc(I,j):

$$simc(i,j)=\frac{\overline{fav(i)}\times\overline{fav(j)}}{\left\|\overline{fav(i)}\right\|\left\|\overline{fav(j)}\right\|} \tag{8}$$

其中: $\overline{fav(i)}$ =[fav(I,1),……,fav(I,k)]。

根据以上公式，结合用户兴趣变化以及类别关联度之后的改进的相似度计算方法为

$$sim_c(i,j)=\beta f(u,i)+(1-\beta)simc(i,j) \tag{9}$$

其中: β 是平衡因子，取值范围为[0,1];

3.1 基于改进相似度的用户聚类协同过滤算法

将以上元素均加入到用户相似度的计算之中后，将对用户进行聚类，最终达到的目的是将相似度较高的用户聚类到一起，从而在进行推荐的时候不会因为采用了兴趣完全不同的用户的数据，对推荐产生偏差，同时也减少了不必要的计算，浪费电脑资源；本文将采用 K-均值聚类方法进行用户聚类，聚类算法使用 Python 语言进行实验操作：

输入：K 个用户分类。

a)从用户集合 U 中随机取 K 个用户，作为 K 个簇各自的中心。clusters=[[random.random()*(ranges[i][1]-ranges[i][0])+ranges[i][0]

for i in range(len(rows[0]))] for j in range(k)]；获取 K 个随机簇中心点。

b)分别计算用户 U(i)与 K 个簇中心的相似度，将 U(i)归到相似度最大的类别。首先给定用户间计算距离即相似度的公式，sim_c(i,j);创建循环，计算比较距离，进行类别选择。

d=distance(clusters[i],row)

if d<distance(clusters[bestmatch],row):

bestmatch=i

bestmatches[bestmatch].append(j);

c)重新计算 K 个簇各自的中心，计算方法是求出所有用户对项目的评分算术平均值，作为簇中心点。

d)对于 U 中所有用户重复步骤 2 和步骤 3 的迭代法更新，直到聚类结果保持不变，则迭代结束，否则继续迭代。

最终获得K个相似项目类别。当用户聚类完成之后，本文需要为某一个用户进行推荐时，只需要在他所属类别中获取与他最相似的的 top—N 用户，根据评分预测公式进行评分，最后为他推荐评分最高的前 K 个项目。

4 实验结果与分析

4.1 实验数据集

本实验采用 MovieLens-1m 数据集， 其中包含了 6640 位用户对 4000 个项目的 100K 个评分， 并将该数 据集的 80%作为训练集， 剩下 20%为测试集。 采用平均绝对误差 MAE（Mean Absolute Error）作为指标，衡量推 荐算法的优劣。 设预测的用户评分集合为{p1，p2，…， pC}，对应的实际用户评分集合为{r1，r2，…，rC}，则平均绝对误差 MAE 定义为：

$$MAE=\frac{1}{n}\sum_{i\in U,j\in I}\left|p_{ij}-r_{ij}\right| \tag{10}$$

4.2 实验结果与分析

4.2.1 α 的选择

α 是用来均衡近期兴趣变化以及远期兴趣变化比重的因子，取值范围为[0,1]，每次增加 0.1，比较 MAE 的变化；从图

中可以看到，α =0.2 时，MAE 最小，推荐效果最佳，这说明用户近期对电影类别的喜爱将对未来一段时间用户选择电影产生巨大的影响。现实生活中也是这样，对电影类别的喜爱具有阶段性，往往一段时间内的兴趣不会有太大的变化。



图 1　α 值对 MAE 值大小的影响

4.3　β 的选择

相似度中聚类信息部分由两部分组成，类别关联度以及用户兴趣时间变化，平衡因子 β 平衡两部分比重，取值范围为[0,1]，每次增加 0.1，α 取 0.1，比较 MAE 的变化；如图 2 所示。

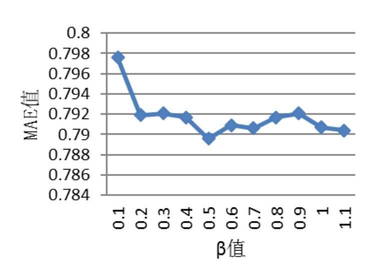


图 2　β 值对 MAE 值大小的影响

从图中可以看出，β =0.5 时，MAE 最小，推荐效果最佳，而 β 值得大小，表明了类别关联度和用户兴趣变化两个因素对用户推荐结果的重要程度，β 取值 0.5，表明两者之间没有明显的偏差，因此关于最后的相似度计算公式，本文取 α =0.2，β =0.5。

4.4　改进的算法性能比较

将基于相似度改进的用户聚类协同过滤与传统协同过滤（基于用户（项目）的协同过滤）以及简单用户聚类算法进行比较分析。简单用户聚类直接进行聚类，基于相似度改进的用户聚类则根据实验结果取 α =0.1，β =0.4，实验结果如图 3 所示。

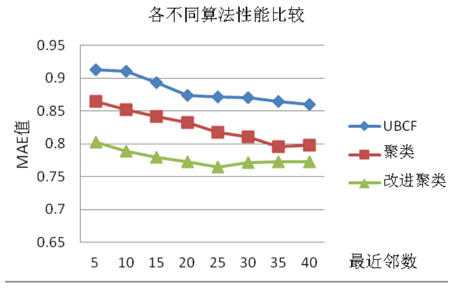


图 3　不同算法性能比较

由实验结果图可知，基于相似度改进之后的聚类算法比传统的协同过滤算法以及传统的聚类算法具有更小的 MAE，并且当最近邻增加到 25 的时候，MAE 最小。实验数据表明，用户的近期兴趣确实对实验结果产生了一定的影响减少了误差，聚类算法的实现，结果优于传统的协同过滤算法。

5　结束语

推荐系统帮助用户解决信息过载问题，已被广泛应用于多个领域 。协同过滤 、基于内容推荐 、基于矩阵推荐和混合推荐是目前较为常见的推荐方法 。本文主要基于用户对电影项目评分以及评分的时间，来进行项目聚类，发现用户的类别喜好度关注度，同时发掘出用户的近期喜好，来改进用户间相似度的计算法则。通过查阅相关资料，在考虑用户近期喜好的同时，也加入了用户以往的兴趣因素。最终实验结果表明，改进后的聚类算法的确误差更小了。推荐算法依然在发展进步中，数据稀疏 、过拟合 、可扩展性和多媒体信息特征提取仍是主要问题。现有的技术和方法都不能从根本上解决这 些问题 。随着应用领域的不断拓展，推荐系统还会面临新的需求与问题 。推荐系统的发展与它面临的问题和 挑战密不可分，针对以上问题的推荐方法研究仍是信息检索 、数据挖掘和机器学习等智能信息处理领域的研究热点 。

参考文献：

[1] 宋瑞平. 混合推荐算法的研究 [D]. 兰州: 兰州大学, 2014.

[2] 王晓军. 利用项目属性和偏好改进协同过滤推荐 [J]. 北京邮电大学学报, 2014, 37 (6): 68-71.

[3] 尹航, 常桂然, 王兴伟. 采用聚类优化的 K 近邻协同过滤算法 [J]. 小型微型计算机系统, 2013, 34 (4): 806-809.

[4] 王明佳, 韩景倜, 韩松乔. 基于模糊聚类的协同过滤算法 [J]. 小型微型计算机系统, 2012, 38 (24): 50-52.

[5] 孙辉, 马跃, 杨海波, 等. 一种相似度改进的用户聚类协同过滤推荐算法 [J]. 小型微型计算机系统, 2014, 35 (9): 1967-1970.

[6] 邢春晓, 高凤荣, 战思南, 等. 适应用户兴趣变化的协同过滤推荐算法 [J]. 计算机研究与发展, 2007, 44 (2): 296-301.

[7] 陈志敏, 李志强. 基于用户特征和项目属性的协同过滤推荐算法 [J]. 计算机应用, 2011, 31 (7): 1748-1751.

[8] Zhang J, Lin Y, Lin M, et al. An effective collaborative filtering algorithm based on user preference clustering [J]. Applied Intelligence, 2016, 45 (2): 230-240.

[9] Li S, Yuan X, Han H. A kind of collaborative filtering algorithm based on user clustering and time stamp [C]// Proc of International Symposium on Advances in Electrical, Electronics and Computer Engineering. 2016.

[10] Chen Q, Li W, Liu J. Collaborative Filtering Algorithm Based on Item Attribute and Time Weight [C]// Proc of International Conference on Automatic Control and Information Engineering. 2016.

[11] Yao Z, Zhang Q. Item-based clustering collaborative filtering algorithm under high-dimensional sparse data [C]// Proc of International Joint

Conference on Computational Sciences and Optimization. 2009: 787-790.

[12] Yu Y, Zhu S, Chen X. Collaborative filtering algorithms based on Kendall correlation in recommender systems [J]. 武汉大学学报: 自然科学英文版, 2016.

[13] Li J, Liu X. An Important Aspect of Big Data: Data Usability [J]. Journal of Computer Research & Development, 2013, 50 (6): 1147-1162.